

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
21 December 2000 (21.12.2000)

PCT

(10) International Publication Number
WO 00/78002 A2

- (51) International Patent Classification⁷: H04L 29/00 (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (21) International Application Number: PCT/US00/15692
- (22) International Filing Date: 7 June 2000 (07.06.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/140,675 11 June 1999 (11.06.1999) US
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (71) Applicant (*for all designated States except US*): MICROSOFT CORPORATION [US/US]; One Microsoft Way, Building 114, Redmond, WA 98052 (US).
- Published:
— Without international search report and to be republished upon receipt of that report.
- (72) Inventor; and
- (75) Inventor/Applicant (*for US only*): MUNDIE, Craig, J. [US/US]; 7009 204th Drive N.E., Redmond, WA 98053 (US).
- (74) Agent: WIGHT, Stephen, A.; Klarquist, Sparkman, Campbell, Leigh & Winston, LLP, One World Trade Center, Suite 1600, 121 S.W. Salmon Street, Portland, OR 97204 (US).
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*



WO 00/78002 A2

(54) Title: MULTI-DIMENSIONAL AUTHORITATIVE NAMES REGISTRY IN PERVASIVE COMPUTING

(57) Abstract: As an aid in locating devices and services, an authority registers names within a set of dimensions, and may charge recurring registration and usage fees. The set of dimensions include devices, services, networks, venues, spaces, people, processes, and events. The names within each dimension can have a hierarchical or non-hierarchical organization. The names may map to an address (e.g., an IP address) to operationally access a protocol to interact with the named item over a global network, such as a COM object, java applet, HTML, XML or other structured data. The authoritative name space created by this registry also operates to define concepts of proximity on the network, including geographical and other proximity concepts of relatedness of items in the various dimensions of the authoritative registry. The names serve as implicit terms in combination with an explicit search query to assess proximity of devices and services available over the global network.

MULTI-DIMENSIONAL AUTHORITATIVE NAMES REGISTRY IN PERVASIVE COMPUTING

TECHNICAL FIELD

5 This invention relates generally to ubiquitous or pervasive computing, and more particularly relates to a device-specific control protocol and proximity networking of such pervasive computing devices.

BACKGROUND

Various trends in the computer, networking and consumer electronics fields are leading towards a world of ubiquitous or pervasive computing and universal
10 connectivity, where all sorts of everyday-use devices will have computing and network connection capabilities. These trends include an explosive growth in personal computer and multiple PC ownership (now over 50% of U.S. homes), availability of broadband “always-on” Internet access (cable, satellite, and ADSL), increasing popularity of mobile personal computing and communications devices (laptop, hand-held and tablet-
15 size computers, and cell phones), introduction of inexpensive home networking alternatives (Ethernet, IEEE 1394, infrared, radio frequency, power line carrier, and phone line-based), transition from analog to digital formats of audio and video content, and a sufficiently low and decreasing cost to embed “intelligence” and “connectivity” in traditionally “dumb” devices. These trends are leading manufacturers to design all
20 varieties of devices (including communications devices, office equipment, audio/video equipment, entertainment devices, heating/ventilation/air-conditioning equipment, security systems, home appliances, and automobiles, among others) to have built-in computing and network connection capabilities, such that interconnected computing devices are truly pervasive in our home, work and public environments. The facile
25 interconnectivity of intelligent devices enables consumers both at home, in the workplace and public places to remotely control devices; to move digital data in the form of audio, video and still images between devices; to share information among

devices and globally with the Internet; and to exchange structured and secure digital data to support electronic commerce, private communications and other functions.

Pervasive computing with universal connectivity creates an environment where a large multitude of devices are globally interconnected and accessible to each other
5 through a global computer network, e.g., the Internet and other networks connected thereto. With this universal connectivity through global networks, devices need no longer attach directly in order to interact with each other. Devices that are geographically remote can interact with each other over a global network independent of the physical distance that separates them. In fact, on global networks, characteristics
10 of physical locality almost cease to exist. On the Internet, information is routed to and from a device through use of an Internet Protocol (IP) address assigned to the device. But, the device's IP address has no necessary correlation to its geographical location. This makes it difficult to deduce the location of a particular device based on the device's IP address.

15 However, for many practical applications of pervasive computing, geographical proximity remains very significant. For example, a sprinkler system can have embedded intelligence and a network connection to vary its watering schedule based on a downloaded weather report, but preferably accesses the weather report for its locale. A driver of a "smart" car can access traffic reports to compute an optimized route to the
20 driver's destination, but preferably obtains a traffic report relevant to its current location. A person on lunch break in an unfamiliar city might desire to look up and order food at a restaurant with a particular ethnic cuisine, and the search preferably would yield nearby restaurants within walking distance. A tourist at an airport preferably accesses airline arrival and departure schedules for that same airport. A
25 presenter in a conference hall most preferably can control and download presentation content to the audio/video equipment in the same conference hall, rather than the one next door or across the country.

With a very large number of globally interconnected devices and no direct indication of geographical locality on a global network, the advent of pervasive

computing creates a problem of being able to easily and quickly locate a particular device or service over the global network. There is a risk that the desired device or service will be indistinguishable from a multitude of other similar devices and services on the global network.

5 For example, Internet users currently navigate to hypertext or "web" pages of interest on the Internet through use of an Internet search engine. The user enters a search query containing one or more key words, or a phrase. The Internet search engine returns a list of links (also called "hits") to web pages related to the key words. Some Internet search engines rely on a catalog of the pages compiled by human critics to
10 determine the relevance of web pages to the keywords of a query. Other search engines employ automated document indexing techniques that extract the unique words encountered on web pages, and measure each web page's relevance by the occurrence(s) on the web page of the keywords specified in the query. In either case, almost any query using the search engine is likely to return hundreds or even thousands
15 of hits. Further, the web page that may be of most interest to the user often is not near the top of the list. Also, many web pages are omitted from analysis by search engines. So, the web page that is of most interest to the user may be outside the search engine's scope.

 For many useful applications, however, the proximity of a device or service
20 accessed via the web can be very significant to the user. If a user wants to order a product (such as a watch) over the Internet for example, the user's primary criteria may be the make, model and price. A search with an Internet search engine for a vendor of the desired make and model will return hits on vendor web pages (and possibly many irrelevant web pages) without regard to their physical location. However, the relative
25 location of the vendor to the user may have unrecognized significance, such as where the list of vendor hits returned in a web search are located in a foreign country which would add to shipping costs and impact time of delivery. At the same time, it is difficult to construct a keyword query to an Internet search engine that will adequately distinguish between web pages based on geographical proximity.

Accordingly, there is a need in a pervasive computing environment for the capability to locate and access devices and services on a computer network based on proximity (sometimes referred to as "proximity networking").

5 A further challenge of pervasive computing is once a device or service is located on the global computer network, how does a client device interact with that device or service when the client device has no a priori knowledge of the device or service and its operational functions. Accordingly, there is also a need for the capability to access device-specific and service-specific operational information in a standard way, by which a client device can control operational functions of the located device or service.

10

SUMMARY

The techniques, systems and processes described herein provide a capability to access device- or service-specific operational information and perform proximity-based networking functions with computing devices using an authoritative names registry having multiple dimensions, such as in a pervasive computing environment. An
15 authority registers names within the various dimensions of the registry, and provides name look-up services. The names within each dimension may be related according to a hierarchical organization, or alternatively in a non-hierarchical manner. A group of names from various dimensions of the registry that are associated with a particular use define a measure of proximity relative to devices and services accessible on a computer
20 network (which preferably is global in scope). This measure of proximity includes geographical locality, but also can encompass other proximity concepts of relatedness of items in the various dimensions of the authoritative names registry. The authoritative multi-dimensional names registry therefore can be used for assessing proximity in proximity networking operations.

25

In an embodiment illustrated herein, the dimensions of the authoritative names registry include devices, services, networks, venues, spaces, people, processes, and events. Names under the devices dimension relate to devices with embedded computing capability that are accessible on the computer network. Names under the

services dimension relate to services provided over the computer network. The networks dimension's names relate to communications and other networks (whether related to the computer network or no), and their constituent parts. Names in the venues and spaces dimensions relate to geographical places or locations, and may also include virtual locations. Venues are locations that comprise aggregations of individual places, such as a sports arena, convention center, mall or the like. Spaces are individually identifiable places. The people dimension's names relate to people and units that group people collectively, such as families and couples. Names in the processes dimension relate to business or other organizational processes or protocols, such as orders, invoices, requests, notices and other forms or protocols with which business and other activities of the organization are conducted. The events dimension's names relate to occasions, happenings or activities in which people participate, such as meetings, conferences, sports games, theatrical presentations, concerts, and the like.

In the illustrated embodiment, a name in a dimension of the authoritative names registry is resolved through the name look-up service of the authority to an address on the computer network where interactive code and/or data can be accessed, such as by use of a computer network browser. For example, the interactive code and/or data can take the form of an executable program (e.g., a COM object, a Java applet or like), or structured data (e.g., an XML or HTML file) which provides a capability to interact over the network with an item (of a type appropriate to the respective registry dimension) that is identified by the name. For example, a name in the devices dimension may resolve to a computer network address where a COM object or XML data can be accessed that control operation of the named device.

The names register is authoritatively administered by an authority, which assigns name registrations within the various dimensions to particular registrants. An authority-managed registry has an important advantage of resolving disputes between registrants over ownership of names in the registry, and thereby avoiding duplicative names usage, misrepresentation and other ambiguities. Authority administration of the names lookup service also aids in preventing fraud.

A devices dimension of the names registry has names organized in a hierarchical arrangement, in which names of generic device types are ordered at top levels and increasingly specific names are registered within the hierarchy under their more generic type names. The authority can maintain one or more levels of generic device type names in the hierarchy, and charge registration and usage fees for more device-specific names registered under these levels. The device specific names preferably resolve to a computer network address where structured data (e.g., an XML file) or an executable file (e.g., a COM object) that defines a protocol for activating operational functions of the device and otherwise to control or interact with the device. Access to the device's protocol through the names registry can be selectively controlled through cryptographic security.

Any names from the authoritative names registry that may be associated with a desired use of devices or services via the computer network can serve as implicit search terms which are processed together with an explicit search query posed by a user to locate the devices or services. For example, the associated names can include a name registered in the people dimension of the authoritative names registry for the user; a name registered in the devices dimension for a device with which the user initiates the search; a name registered in the events dimension for a conference currently attended by the user; a name registered in the venues or spaces dimension for the user's current location; and so on for the various dimensions of the authoritative names registry. These implicit search terms are processed to assess the proximity of items identified in the explicit search query, and present the items appropriately based on proximity. Such proximity-based searches with proximity-defining registered names as implicit search terms also can be posed to locate items other than devices and services on the computer network, such as items identified by names in the various dimensions of the authoritative names registry or other information.

Names associated to a desired use also can serve as implicit search terms of an inverse search, which delivers an alert or notification of an item meeting the search's parameters including proximity defined by the implicit search terms. For example, a

user can pose an inverse search for a person at a conference, which alerts the user when the person arrives at the conference.

The authority manages the authoritative names registry as a business process. The authority can charge recurring fees for registration of names in the various
5 dimensions of the authoritative names registry, as well as fees for usage of the authoritative names registry in either a name look-up service operation or a proximity-based search. The authority can charge the usage fee to the user, or alternatively to the owner of a name registration. Where names are organized hierarchically in a dimension
10 of the authoritative names registry, the authority can charge the registration fees for names at one or more levels of depth in the hierarchy.

Additional features and advantages will be made apparent from the following detailed description of the illustrated embodiment which proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

15 Figure 1 is a block diagram of a computer system that may be used to implement an illustrated embodiment for device control protocol access and proximity networking using a multiple dimensioned authoritative names registry.

Figure 2 is a block diagram of a device having embedded computing and networking capability per universal-plug-and-play (UPNP) standards that may be used
20 in combination with the computer system of Figure 1 in the illustrated embodiment for device control protocol access and proximity networking using a multiple dimensioned authoritative names registry.

Figure 3 is a block diagram of a software architecture per UPNP standards in the embedded computing device of Figure 2

25 Figure 4 is a data flow diagram of a process for automatic network introduction of the embedded computing device of Figure 2 into an ad hoc computer network environment per the UPNP protocol.

Figure 5 is a data flow diagram of a process for automatic network introduction of the embedded computing device of Figure 2 into a configured computer network environment per the UPNP protocol.

Figure 6 is a block diagram of a software architecture of a client device per
5 UPNP standards having embedded computing and networking capability that may be used in the illustrated embodiment for device control protocol access and proximity networking using a multiple dimensioned authoritative names registry.

Figure 7 is a block diagram of an exemplary home or office pervasive
10 computing environment having a variety of computers as per Figure 1 and embedded computing devices as per Figure 2 interconnected per UPNP standards that may be used in the illustrated embodiment for device control protocol access and proximity networking using a multiple dimensioned authoritative names registry.

Figure 8 is a diagram of computers in an illustrated implementation of the
15 invention that provide an authoritative names registry and proximity-based search engine for use in device control protocol access and proximity networking by a client device to locate and access devices and services over a computer network.

Figure 9 is a listing of dimensions in the authoritative names registry of Figure
8.

Figure 10 is a listing of a name registered under a dimension of the authoritative
20 names registry of Figure 8.

Figure 11 is a diagram of a hierarchical organization of names in a dimension of the authoritative names registry of Figure 8.

Figure 12 is a view of a user interface screen display on the client device of
Figure 8, with which a user can initiate a proximity-based search.

25

DETAILED DESCRIPTION

The following discussion is directed toward device control protocol access and proximity networking using an authoritative names registry having multiple dimensions. In one embodiment illustrated herein, the authoritative names registry is operated within

a large computer network 308 (Figure 8) via which a multitude of computers and embedded computing devices are interconnected. Preferably, this computer network is global in scope, such as the Internet. Alternatively, the authoritative names registry can be employed within other networking configurations, such as a local area network, an intranet or other computer network.

Exemplary Computer

Figure 1 and the following discussion are intended to provide a brief, general description of a suitable computer which may be used in the illustrated implementation of the invention. While the invention will be described in the general context of computer-executable instructions of a computer program that runs on a computer, those skilled in the art will recognize that the invention also may be implemented in combination with other program modules. Generally, program modules include routines, programs, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The illustrated embodiment of the invention also is practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to Figure 1, an exemplary system for implementing the invention includes a conventional computer 20 (such as personal computers, laptops, palmtops or handheld-PCs, set-tops, servers, mainframes, and other variety computers), including a processing unit 21, a system memory 22, and a system bus 23 that couples various system components including the system memory to the processing unit 21. The processing unit may be any of various commercially available processors, including Intel x86, Pentium and compatible microprocessors from Intel and others, including

Cyrix, AMD and Nexgen; Alpha from Digital; MIPS from MIPS Technology, NEC, IDT, Siemens, and others; and the PowerPC from IBM and Motorola. Dual microprocessors and other multi-processor architectures also can be used as the processing unit 21.

5 The system bus may be any of several types of bus structure including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of conventional bus architectures such as PCI, VESA, AGP, Microchannel, ISA and EISA, to name a few. The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS), containing the basic
10 routines that help to transfer information between elements within the computer 20, such as during start-up, is stored in ROM 24.

 The computer 20 further includes a hard disk drive 27, a magnetic disk drive 28, e.g., to read from or write to a removable disk 29, and an optical disk drive 30, e.g., for reading a CD-ROM disk 31 or to read from or write to other optical media. The hard
15 disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions, etc. for the computer 20. Although the description of
20 computer-readable media above refers to a hard disk, a removable magnetic disk and a CD, it should be appreciated by those skilled in the art that other types of media which are readable by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, and the like, may also be used in the exemplary operating environment.

25 A number of program modules may be stored in the drives and RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38.

 A user may enter commands and information into the computer 20 through a keyboard 40 and pointing device, such as a mouse 42. Other input devices (not shown)

may include a microphone, joystick, game pad, satellite dish, scanner, or the like.

These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor
5 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, computers typically include other peripheral output devices (not shown), such as speakers and printers.

The computer 20 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. The remote computer
10 49 may be a server, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 20, although only a memory storage device 50 has been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area network (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices,
15 enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 20 is connected to the local network 51 through a network interface or adapter 53. When used in a WAN networking environment, the computer 20 typically includes a modem 54 or other means for establishing communications (e.g., via the LAN 51 and a gateway or proxy
20 server 55) over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the computer 20, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other
25 means of establishing a communications link between the computers may be used.

In accordance with the practices of persons skilled in the art of computer programming, the present invention is described below with reference to acts and symbolic representations of operations that are performed by the computer 20, unless indicated otherwise. Such acts and operations are sometimes referred to as being

computer-executed. It will be appreciated that the acts and symbolically represented operations include the manipulation by the processing unit 21 of electrical signals representing data bits which causes a resulting transformation or reduction of the electrical signal representation, and the maintenance of data bits at memory locations in the memory system (including the system memory 22, hard drive 27, floppy disks 29, and CD-ROM 31) to thereby reconfigure or otherwise alter the computer system's operation, as well as other processing of signals. The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, or optical properties corresponding to the data bits.

10 Exemplary Embedded Computing Device

Figures 2 and 3 are intended to provide a brief, general description of a suitable embedded computing device 100 which may be used in the illustrated implementation of the invention. The embedded computing device 100 can be any variety of device incorporating electronics to control operational functions (operational circuitry 106), and in which computing and networking capabilities are embedded. For example, devices in which computing and networking functions can be embedded include communications devices (e.g., telephones, cell phones, audio and video conferencing systems, 2-way radios, etc.), office equipment (printers, fax machines, copiers, dictation, etc.), audio-video equipment (audio and video recorders and players, including televisions, radio receivers, compact disk (CD), digital video disk (DVD), camcorders, etc.), entertainment devices (set-top boxes, game consoles, etc.), environment control equipment (thermostats, heating/ventilation/air-conditioning equipment, light switches, etc.), security systems, home appliances (coffee makers, dishwashers, clothes washer/dryer), automobiles, public facilities equipment (signs, traffic signals, etc.), manufacturing equipment, and many others.

With reference to Figure 2, the device 100 includes a processing unit 102, and a memory 104 to provide embedded computing capability. The processing unit 102 has hardware interfaces to the operational circuitry 106 that operates devices functions. The processing unit 102 can be a microprocessor or micro-controller, such as are available

from Intel, Motorola, IBM, and others. The memory 104 preferably incorporates RAM and ROM to hold software and data for basic operating code as well as for user applications.

5 The device 100 also includes a network adapter 108 for connecting with a network media 110 that is interconnected with the computer network in which the authoritative names registry (described below) is implemented in accordance with the invention. The network adapter 108 can be a network interface card (or chip set integrated on a single board with the processing unit 102) appropriate to the particular network media 110. The network media can be any of various wired or wireless
10 network media, including Ethernet, IEEE 1394 (a.k.a. firewire), radio frequency (including satellite, cell, pager, commercial signal sideband, etc.), power line carrier (PLC), phone line, and television cable, among others.

With reference now to Figure 3, the embedded computing device 100 (Figure 2) has a software architecture 120 that conforms to a proposed Universal Plug And Play (UPNP) standard. UPNP is a proposed device configuration protocol for
15 interconnecting embedded computing devices over a computer network, that features automatic networking, scalability, and adherence to open industry standards. For more information on UPNP, see, Christensson, Larsson, "Universal Plug And Play Connects Smart Devices," WinHEC 99 White Paper; and "Universal Plug And Play: Background" available from the Universal Plug and Play (UPNP) Forum.

UPNP provides a mechanism for the embedded computing device to operate in the Internet, as well as networks that have no administrator and no connection to the Internet, and hence no access to configuration services like the Dynamic Host Configuration Protocol (DHCP). DHCP is a mechanism for providing devices with
25 configuration information needed to access the Internet. The mechanism functions through the use of a multicast request for configuration information that is generally responded to with an IP address and DNS server location. Additional information can only be returned in the response. [For more information, see R. Droms, Dynamic Host Configuration Protocol, IETF RFC 2131 (March 1997)].

In non-configured (ad-hoc) networks, UPNP uses the AutoIP protocol. AutoIP is an enhancement to DHCP that allows devices to claim IP addresses in the absence of a DHCP server or similar IP configuration authority. IP addresses are claimed from a reserved range that is not allowed to be transmitted on the open Internet; thus they are only good for the local network. The embedded computing device 100 claims an address by randomly generating an address in the reserved range and then making an ARP request to see if anyone else has already claimed that address. AutoIP systems will continually check for the presence of a DHCP server so that if one should ever come online, all the AutoIP devices will attempt to switch their IP addresses to one provided by the DHCP server. This allows a network to operate in isolation, be connected to the Internet with DHCP support and then to be returned to isolation. This type of scenario will be common in homes that use dial-up access.

The UPNP protocol also uses Multicast DNS for addressing the embedded computing device 100. The Internet Domain Name System (DNS) is a mapping system that translates human readable domain names, like microsoft.com, into their equivalent IP address. Most corporate intranets implement an internal version of the same technology to provide the same services. In small networks, such as at home or in small business, DNS servers may not exist. Multicast DNS allows DNS requests to be multicast. This allows a machine to see requests for its own name and respond to them. Like AutoIP, Multicast DNS is only used when a DNS server is not available. (For more information, see B. Woodcock, Zocolo, and B. Manning, Multicast Discovery of DNS Services, IETF Internet Draft, "draft-manning-multicast-dns-01.txt.")

UPNP implements a peer discovery mechanism that uses the Simple Service Discovery Protocol (SSDP) for discovery of devices on IP networks. SSDP is based on profiles. A single identifier specifies a profile that defines a contract between the client and service (e.g., operational functions provided by the embedded computing device). By identifying itself with the profile, the service advertises compliance with the associated contract.

Using a single identifier makes it possible to implement an extremely simple discovery system. Clients send out a User Datagram Protocol (UDP) multicast packet containing the identifier of the desired service on some standard channel. Services listen on the standard channel, read the request, see whether they provide the service, and respond if so.

UPNP also provides a Directories mechanism to allow discovery to scale - to the entire Internet if needed. When present, a directory will read all incoming service requests and respond to them itself. This requires that all services (e.g., the embedded computing device 100) register with the directory so that the directory is able to properly answer on their behalf. The directory is also responsible for communicating with other directories in order to determine whether the service is available within the local network, the WAN and potentially the Internet.

To simplify the discovery protocol, directories are treated as proxies. A proxy is a service that accepts requests and takes responsibility for finding the proper response. When a client comes online, it will perform discovery for the proxy. If the proxy is present, then the client will send all future discovery requests to the proxy. If the proxy isn't present, then the client will send all discovery requests to the reserved discovery multicast channel. Regardless of the presence of a proxy, the client's request format and procedures will always be the same. The only difference will be the address to which the client sends its requests. For services, the difference between a proxied and unproxied network is their need to answer discovery requests. On a proxied network, services need do nothing once they have registered with the proxy. On an unproxied network, they answer discovery requests directly.

SSDP uses the UDP- and Transmission Control Protocol (TCP)-based Hypertext Transport Protocol (HTTP) to provide for service discovery. SSDP uses a Uniform Resource Identifier (URI) to represent the service and the OPTIONS method to provide for discovery. SSDP also will provide support for proxies. These proxies, which are really just fronts for directories, redirect discovery requests to themselves. It is the proxy's job to collect announce requests in order to determine what services are

available as well as to communicate with other proxies in order to provide for scalable service discovery.

The discovery process returns only the basic information needed to connect to the embedded computing device. Once a service has discovered its peers, the service
5 often needs to find out more information in order to work best with them. The description process returns a schema providing descriptive data about the service.

A schema is a structured data definition that defines a set of structured values that provide descriptive information about a service. UPNP uses the Extensible Markup Language (XML) for schema, because XML's self-describing structured data format
10 provides the level of expressiveness and extensibility needed by a universal schema and data format.

Accordingly, UPNP supports automatic network introduction, meaning that devices and their related services have the ability to be self-describing and allow automatic configuration. When a device is plugged into the computer network, the
15 device automatically configures itself and acquires a TCP/IP address. The device then announces its presence to other devices already on the network using a simple discovery protocol based on the Internet HTTP protocol and is immediately ready to share its services with any device that requests them.

With UPNP, device developers are not required to develop specific device
20 drivers to operate under UPNP. The task of preparing a device for operation in this network environment thus is fairly simple. Moreover, in configured networks, dynamic detection allows an operating system to immediately begin using added devices or stop using removed devices without rebooting.

UPNP Devices support automatic discovery, identification, and configuration to
25 achieve interoperability in the home environment, but must also operate correctly in a managed corporate network. Devices can be networked instead of being attached directly to a PC, and devices are all autonomous citizens on the network, able to talk with each other and exchange information. UPNP provides a unified way of performing directory services with automatic configuration. Capability for simple discovery

mechanism used in the home environment provides the ability for any device to become a node on the global Internet. Additionally, directory services can be leveraged if they are available in the corporate environment.

UPNP provides a common set of interfaces for accessing devices and services,
5 enabling the operational unification of diverse media types. Communications protocols for Universal Plug and Play are based on industry standards, especially key Internet standards such as TCP/IP, HTML, XML, HTTP, DNS, LDAP, and others. Individual implementations for particular networks and buses are built on established protocols.

As shown in Figure 3, the software architecture 120 of the embedded computing
10 device 100 (Figure 2) includes the following software code modules that implement UPNP: device functions 122, simple discovery 124, Hypertext Transport Protocol (HTTP) 125, Transmission Control Protocol/Internet Protocol (TCP/IP) stack 126, Autonet 128, Dynamic Host Configuration Protocol (DHCP) 130, and physical media 110 (also shown in Figure 2). The device functions 122 is a software code module to
15 implement the device's functionality. For example, where the embedded computing device is a VCR, the device functions code can include code to implement start, stop, pause, record and other functions that the VCR can perform.

The simple discovery 124 is a software code module (about 4 Kbytes) that implements a simple discovery procedure (described below) for automatic network
20 introduction under the UPNP protocol.

The simple discovery procedure additionally provides an Extensible Markup Language (XML) format device description, which is downloaded to clients that access the device to allow activation of device functionality from the client. XML is a textual, tag-based markup language. It was originally designed to be the "webby" simplification
25 of SGML (Standard Generalized Markup Language), and is therefore intended to be used to create "vocabularies" of tags that can be used to apply semantic markup to documents, such as who the author was, what constitutes a paragraph (semantically, not from a display point of view), when the author last had breakfast, and so on (For more information, see Layman et al., "XML-Data," W3C Note 05 Jan 1998). In the context of

UPNP, XML is used to provide the description of services and capabilities of the embedded computing device. The embedded computing device makes its features visible to clients by providing its XML device description, which the client can use to activate device functions 122. For example, if the device is a camera, the client's
5 browser can direct the camera to zoom in/out or adjust contrast using the mechanism of XML.

The XML device description can provide links (via a uniform resource locator or URL address) to an accompanying XSL format style sheet. The XSL style sheets are used to present the data in different ways, i.e., the style sheets are applied to present
10 different views of the same data. For example, if the device contains a file system, one style sheet can show the file selections; another shows the file sizes in some sort of diagram; yet another style sheet could make thumbnails of these image files.

The HTTP 125 is a software code modules (about 20 Kbytes) that implements the standard HTTP protocol, which is an open standard mechanism for client/server
15 message-based communication. HTTP provides for proxying, content negotiation and security. [For more information, see R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, Hypertext Transfer Protocol-HTTP/1.1, IETF RFC 2068 (January 1997)].

The TCP/IP stack 126 implements the standard TCP/IP networking protocols for communication on the computer network. The Internet Protocol (IP) is the foundation
20 protocol of the Internet. It defines how a single message is sent from a source through zero or more routers to its final destination. It covers issues such as message length, message fragmentation, addressing, and routing concerns. [For more information, see J. Postel (Editor), Internet Protocol, IETF RFC 791 (September 1981)]. The Transmission Control Protocol (TCP) is an IP-based protocol that provides support for the reliable,
25 ordered delivery of messages over IP. [For more information, see J. Postel (Editor), Transmission Control Protocol, IETF RFC 793 (September 1981)]. Additionally, User Datagram Protocol (UDP) and Internet Group Management Protocol (IGMP) multicast send/listen capability are included in the implementation.

The Autonet 128 is a software code module also used for automatic network introduction via AutoIP in the UPNP protocol. Autonet uses a predefined set of IP addresses and, when a device is connected to the network, it pings an address in this address space. If it gets no replies, the device assumes that the address is available and
5 assigns it to itself. To make this functionality even more useful it is combined with Multicast DNS, in which the device itself holds its own name. Thus it is not even necessary to determine what IP address the device assigned to itself, because its name can always be used instead. An IP Multicast is a mechanism for sending a single message to multiple recipients. IP multicasting is especially useful for discovery
10 operations where one does not know exactly who has the information one seeks. In such cases, one can send a request to a reserved IP multicast address. Any services that can provide the requested information will also subscribe to the multicast request and thus be able to hear the information request and properly respond. [For more information, see S. Deering, Host Extensions for IP Multicasting, IETF RFC 1112
15 (August 1989)]. Multicast DNS is a proposal to the IETF on rules for making normal DNS requests using multicast UDP. [For more information, see B. Woodcock, B. Manning, Multicast Discovery of DNS Services, IETF Internet Draft, "draft-manning-multicast-dns-01.txt".]

The DHCP 130 is a software code module that implements the Dynamic Host
20 Configuration Protocol (DHCP), which is a mechanism for providing devices with configuration information needed to access the Internet. The mechanism functions through the use of a multicast request for configuration information that is generally responded to with an IP address and DNS server location. Additional information can only be returned in the response. [For more information, see R. Duoms, Dynamic Host
25 Configuration Protocol, IETF RFC 2131 (March 1997)].

Figures 4 and 5 show processes 134, 140 per the UPNP protocol for automatic network introduction of the embedded computing device 100 (Figure 2) into an ad hoc (where the device does not have a configured IP address) and a configured computer network environment, respectively. The automatic network introduction process

establishes an appropriate configuration (e.g., with an IP address) of the embedded computing device upon connection to a server computer 136 on the computer network 308 (Figure 8), so as to enable access to the device from a client. The processes 134, 140 involve five phases: announce, discovery, response to discovery, autonet, and
5 device description.

At the announce phase, the embedded computing device 100 sends out a small multicast packet so that other devices can find it on the network. The multicast message packet essentially says, "I am here, I am, (say), a camera, and you can reach me at this IP address or URL."

10 At the discovery phase, the embedded computing device 100 listens for a discovery packet coming from a simple discovery client, i.e., the device announces itself, then listens for discovery. The discovery packet also is sent out by multicast.

At response to discovery, the embedded computing device 100 listens to the multicast address and then parses the information from a Simple Discovery request to
15 decide if the request is for its kind of device. If so, the device 100 then sends back a response packet containing the following information: the IP address or URL where it can be reached; identification of its own device type; and the discovery packet ID so the requesting client knows which request is being answered.

At the Autonet phase, the Autonet module 128 of the embedded computing
20 device 100 uses a predefined set of IP addresses and, when the device is connected to the network, it pings an address in this address space. If no reply is received, the device 100 assumes that the address is available and assigns it to itself. Alternatively, the device 100 may combine Autonet with Multicast DNS, and itself hold its own name. In which case, it is not necessary to determine what IP address the device assigned to
25 itself, because its name can always be used instead.

Both the Announce and Discovery packets also contain a link or a URL to an XML file that is used by the embedded computing device at the device description phase to describe itself (i.e., its functionality). This XML data contains all the facts about the device: XML can also have URLs that point to appropriate style sheets (XSL

files) that are used for optimal presentation. The XSL style sheets are used to present the data in different ways, i.e., the style sheets are applied to present different views of the same data. For example, if the device contains a file system, one style sheet can show the file selections; another shows the file sizes in some sort of diagram; yet
5 another style sheet could make thumbnails of these image files.

Exemplary Client

With reference now to Figure 6, a client 302 (Figure 8) that accesses and uses the embedded computing device 100 over the computer network has an exemplary client software architecture 150, which includes software code modules for applications
10 152, simple discovery 154, XML 155, LDAP 156, TCP/IP stack 158 and a network interface card (NIC) 160 that provides a physical connection to the computer network 308 (Figure 8). The applications 152 is a software code module that provides a user interface features (described below) with which the user can initiate a proximity-based search (described more fully below) for locating desired devices (e.g., embedded
15 computing device 100) and services on the computer network, and also user interface features to interact with the located device or service. The applications 152 can include an Internet browser, such as the Microsoft Internet Explorer, that can present the XML device description in accordance with an associated XSL style sheet for interaction with the embedded computing device and activation of its operational functionality.

20 The simple discovery 154 is a software module that implements the above-described simple discovery per the UPNP protocol. The XML 155 is a software module that processes the XML device description and XSL style sheets for presentation in the application's user interface. The LDAP 156 implements the standard LDAP directory protocol for name look-up. The TCP/IP stack 158 implements the TCP/IP protocol for
25 communications over the computer network 308 (Figure 8).

Illustrative Pervasive Computing Environment

Figure 7 illustrates a pervasive computing environment 200, such as may be installed in a home, office or public place, which includes a large number of embedded computing devices, such as the illustrated device 100 (Figure 2). The pervasive

computing environment 200 includes personal computers 202, 204 (e.g., of the type shown in Figure 1) connected via a local area network (LAN) 206. The PC 202 is connected via a universal serial bus 216 to a telephone modem 210, XDSL interface 211 or a cable modem 212, which in turn provide a connection with the computer network 308 (Figure 8), e.g., the Internet.

Various embedded computing devices also connect to the computer network 308 via various network connections to the PCs 202, 204. These include an audio device 214 (e.g., speakers, radio tuner, microphone), and printer 215 which connect to the PC 204 through a USB 217. Also, a digital camera 220, a handheld PC (H/PC) 221 and another personal computing device 222 connect via an infrared port (IRDA) 224, which also attaches to the PC 204 through the USB 217. Also, lighting switches 230 and like home appliances are connected via an A/C power line-based networking 232 to the PC 202. Further, a chain of IEEE 1394 cables 248 connect a digital TV 240, DVD player 241, digital video camcorder (DV/DVC) 242, an audio device 243 (e.g., CD player/recorder, radio receiver, amplifier, and like audio system component), and a game console 244. Devices, such as a portable telephone 250 and remote control 251, have a radio frequency network connection with the PC 204.

With their various inter-networked connections, the embedded computing devices 202, 204, 210-212, 214-215, 220-222, 230, 240-244, and 250-251 are "visible" and accessible from a client device 150 (Figure 7) also connected to the computer network 308.

Authoritative Multi-Dimensional Names Registry

With reference now to Figure 8, the illustrated implementation of the invention has an authoritative names registry 310 and a proximity-based search engine 312 that provide device control and proximity-based networking within an operating environment 300 that includes a client device 302 and a variety of devices 304 and services 305 accessible to the client device 302 via a computer network 308 (e.g., the Internet and its associated computer networks, or other computer network with large number of interconnected computing devices). The client device 302 can have the

UPNP client software architecture 150 of Figure 6. The devices 304 and services 305 can include embedded computing devices (such as the embedded computing device 100 of Figure 2 and devices shown in the illustrated pervasive computing environment 200 of Figure 7) and their services, that are exposed through the UPNP protocol over the computer network 308. The services 305 also can include any variety of services that can be made available over the computer network, including services to provide access to information, on-line shopping, communications, entertainment and etc.

The authoritative names registry 310 and the proximity-based search engine 312 also reside on computers that are connected to the computer network 308. Although illustrated in Figure 8 as residing on a single computer each, the authoritative names registry 310 and proximity-based search engine preferably are each deployed over a server cluster, group of mirrored server computers, or otherwise distributed over a group of computers in order to service a heavy load of service requests from large numbers of users.

The authoritative names registry 310 contains a database that represents a registry of names, which is used in the illustrated implementation of the invention to provide access to device-control and other protocols and define proximity of the devices 304 and services 305 to the client device 302 and its user. This database can further represent a mapping of individual of the names to an address (e.g., an IP address), that can be used to access a resource over the computer network. This resource preferably is a data or executable file, such as an XML format file (possibly with associated XSL style sheet), a COM object, an HTML format file, a Java applet, or like resource, that describes and/or interacts with an item denoted by the name. In other words, the structured data or executable code defines a protocol for interaction with the device and activation of the device's operational functions. However, the names of the registry 310 need not map to a resource in all cases. Access to the device protocol can be controlled using cryptographic security, such as well-known access control techniques based on private/public key cryptography (e.g., RSA, PGP or like cryptography).

The authoritative names registry 310 also implements a name look-up service. Given a name, this name look-up service determines whether the name is indeed registered in the authoritative names registry 310, and resolves the name to its mapped address on the computer network 308 (if mapped to an address). Preferably, the name look-up service conforms to the Internet standard domain name service (DNS) for look-up of an Internet domain name. A client name look-up request is made using the UDP or TCP protocol to a names registry server. The names registry servers of the authoritative names registry 310 generally are organized in a hierarchy to maintain separate portions of the names in the authoritative names registry. The client's name look-up request can be passed upwards within this hierarchy until reaching the names registry server pertinent to the name look-up request. [For more information on DNS, see C. Partridge, Mail Routing And the Domain System, IETF, RFC 974 (January 1986)].

With reference now to Figure 9, the authoritative names registry 310 has a number of dimensions 350 in which names are organized. These dimensions include devices, services, networks, venues, spaces, people, processes, and events. The organization of names within these dimensions establishes a taxonomy or categorization of items denoted by the names, so as to represent relationships between items according to these dimensions. More specifically, these dimensions conceptually form a multiple dimensional proximity space, which can be used to measure and assess proximity of the devices 304 and services 305 in a search. Each of the names dimensions forms a conceptual axis in this multiple dimensional space. Thus, each of devices, services, networks, venues, spaces, people, processes, and events is an axis of the conceptual proximity space. A set of related names from each of the dimensions designates a point in the conceptual proximity space. With two such sets of names, a proximity distance can be calculated in the conceptual proximity space (e.g., as the square root of a sum of the squares of the differences between the sets in each dimension). In practice, a proximity distance can be calculated between sets that don't include a name in each dimension. The proximity distance in the multiple-dimensional, conceptual proximity

space provides a measure of proximity for proximity networking operations in the illustrated implementation of the invention.

5 The particular names dimensions of the illustrated implementation are chosen to represent factors found to be highly relevant to measuring proximity of the devices 304 and services 305 in a pervasive networking, such as the pervasive networking environment 200 of Figure 7. These factors include factors closely related to geographical proximity, such as the venues and spaces dimensions. But, the chosen names dimensions also represent non-geographical factors such as relationships of devices, services, networks, people, processes and events. Further, even the venues and spaces represent more than simply geographical distance, but also relate to conceptual organizations of geographical space. For example, a room in a building and a sidewalk outside the building may be separated by only a few feet of geographical distance. However, the names denoting these locations in the spaces dimension of the authoritative names registry can be organized (such as by a hierarchical structuring of names in the spaces dimension) to reflect that the room is conceptually more related (proximate) to other rooms in the building than to the geographically nearby sidewalk outside.

10 More particularly, the names in the devices dimension of the authoritative names registry 310 denote devices with embedded computing capability that are accessible on the computer network, such as the embedded computing device 100 or the various devices described above in the pervasive computing environment 200 of Figure 7. The names in the services dimension pertain to services that are available over the computer network 308, such as information services, on-line shopping and other commercial services, entertainment, and the like, as well as individual operations that may be performed by an embedded computing device that is connected to the computer network 308. The names in the networks dimension relate to networks (interconnected group or systems), such as computer networks, communications networks, distribution networks, transportation networks, etc., as well as their constituent parts. The names in the venues and spaces dimensions relate to geographical places or locations, and may also include

virtual locations. Venues are locations that comprise aggregations of individual places, such as a sports arena, convention center, mall or the like. Spaces are individually identifiable places, that may or may not form part of a venue. The names of the people dimension relate to people and units that group people collectively, such as families, couples or other organizations. The names in the processes dimension relate to business or other organizational processes or protocols, such as orders, invoices, requests, notices and other forms or protocols with which business and other activities of the organization are conducted. The events dimension's names relate to occasions, happenings or activities in which people participate, such as meetings, conferences, sports games, theatrical presentations, concerts, and the like, which generally have an associated time and place of occurrence (such as may be denoted in the venues or spaces dimension). Although the dimensions are designated in the illustrated implementation with the terms, "devices, services, networks, venues, spaces, people, processes, and events," alternative implementations can designate these conceptual dimensions using other terms (e.g., the term "locations" or "places" instead of "spaces"). Alternative implementations also can include fewer or additional dimensions than those described for the illustrated implementation.

With reference now to Figure 10, an exemplary name 360 in the devices dimension consists of a sequence of words 361-366 separated by periods as delimiters. The words are organized right to left in order of increasing specificity. More particularly, the right-most word is the most general and denotes the devices dimension. Words progressing to the left reflect narrower sub-categories under the broad devices dimension. Many such sub-categories can exist under each dimension of the authoritative names registry 310 (Figure 8).

With reference to Figure 11, the structure of words that form the exemplary name 360 (Figure 10) reflect a hierarchical organization 380 of names within the device dimension. Each of the words 361-366 corresponds to a node 381-386 in this hierarchical organization 380. The hierarchical organization 380 groups these nodes in successive levels 391-396 which correspond to the successively more specific sub-

categorization within the devices dimension. In the illustrated implementation, the initial level or levels of the devices dimension hierarchy define generic types of devices. For example, the names “camcorder” and “digital-video” at levels 392 and 393 designate generic types of an embedded computing device. Examples of additional generic device types that may be registered at the initial level 392 under the devices dimension include camera, speaker, set-top, VCR, coffee-maker, clock, telephone, and other generic device types of the many varieties of embedded computing devices in a pervasive computing environment, such as the environment 200 of Figure 7. At levels beneath a generic device type name in the hierarchy 380, device specific-names can be registered, such as names of a particular make or manufacturer, model, sub-model, and even serial number or serial number range.

The names within the dimensions of the authoritative names registry 310 (Figure 8) can be organized hierarchically as shown in Figures 10 and 11. Alternatively, the names in a dimension of the registry can have a non-hierarchical organization, e.g., simple flat list, directed graph, or others.

With reference again to Figure 8, the proximity-based search engine 312 provides a search service, which uses the authoritative names registry 310 to assess proximity of the devices 304 and services 305 relative to the client device 302. In the illustrated implementation, the proximity-based search engine uses a set of names associated with the client device 302 as implicit search terms to qualify an explicit search query submitted from the client device 302. These implicit search terms can include a name for the client device 302 within the devices dimension, a name of the client device’s user in the people dimension, a name in the event dimension of an event at which the client device is used, a name in the venues and spaces dimensions of the client device’s location, etc. As described above, such a set of names designates a point in the conceptual proximity space.

Figure 12 shows an exemplary user interface display on the client device 302 (Figure 8) with which its user submits a search query to the proximity-based search engine 312. The user chooses a proximity-based search by selecting a radio button 402

marked "neighborhood." The user also enters explicit search terms in a text box 404, and then clicks on a button 406 labeled "go." This submits a search query that incorporates the text in the text box 404 as explicit search terms, which also is qualified by the set of names from the authoritative names registry 310 that are associated with
5 the client device 302 (or use thereof).

With reference again to Figure 8, the devices 304 and services 305 preferably each include an XML description file. The XML description file is a structured file having parts designated by tags. These tags are used to designate attributes of the device 304 or service 305 described. In the illustrated implementation, the attributes
10 can include names in the authoritative names registry. The names designated in the XML description file of a device or service also form a set to designate a point in the conceptual proximity space represented by the authoritative names registry. The proximity-based search engine 312 calculates the relative distance of this point from the point designated by the client device's names set in the multiple dimensioned,
15 conceptual proximity space to produce a measure of the proximity of the device or service from the client device 302.

The proximity-based search engine 312 uses the proximity measure to qualify the user's explicit search query. For the explicit search terms of the query, the proximity-based search engine 312 employs conventional search techniques, such as
20 may be based on automated indexing of unique words encountered in the device or service's XML description file, or manual editor categorization of the devices 304 and services 305. The proximity-based search engine 312 can apply the proximity measure to pare down the results of the explicit search query, such as to exclude results not within a proximity threshold. Alternatively, the proximity-based search engine 312 can
25 apply the proximity measure to order the results of the explicit search query by proximity, so that the most proximate results are presented at higher priority.

The set of names for the client device 302 also can be derived from an XML description file of the client device 302. This approach works well for attributes of the devices that tend to remain static, such as the name of the client device 302 itself.

However, other attributes such as for the venues, spaces, events and even people can change. For these dimensions, the illustrated implementation can look up an XML description related to known names, such as using the name look-up service of the authoritative names registry. For example, the client device 302 may first determine the
5 name in the authoritative name registry of the user from a user attribute in its own XML description file (which may change depending on who is logged onto the client device). The client device 302 then uses the user's name to retrieve an XML description file (or other information) of the user. This information for user may include a calendar that indicates an events dimension name, venues dimension name and spaces dimension
10 name for an event that the user is scheduled to attend. The client device 302 can then include these names as implicit search terms when the user submits an explicit search query to the proximity-based search engine 312.

Additionally, the proximity-based search engine 312 supports a proximity-based inverse search. In the user interface 400 of Figure 12, the user can designate such
15 inverse search by selecting the radio button 408 labeled "inverse search." In the inverse search, the proximity-based search engine 312 monitors the proximity of the devices 304 and services 305 that meet the explicit search terms, and generates an alert or notification to the client device when any such devices come within the proximity of the client device. This can be used at a convention, for example, to receive an alert when a
20 particular person (who is wearing a name badge embedded computing device) enters a particular meeting or convention center room.

Referring still to Figure 8, the authoritative names registry 310 is managed by an authority, which is a trusted business or organization accountable to the users of the registry and search engine service. The authority performs the necessary function of
25 resolving disputes in ownership of names in the authoritative names registry, and secures the authoritative names registry from unauthorized alteration, fraud or other possible attacks. The authority also administers registration of names in the various dimensions, and preferably operates the authoritative names registry to maintain essentially continuous operation. In the illustrated implementation, the authority

administers at least registration of names at a first level below each dimension (e.g., the level 392 of Figure 11). Names at levels under a registered first level name in the hierarchy can be administered, in turn, by the registered owner of the name, such as by providing an appropriate name look-up service.

5 The authority in the illustrated implementation manages the authoritative names registry 310 as a business process. In this business process, the authority charges recurring registration fees to register a name in a dimension of the authoritative names registry. The authority can charge such registration fees for one or more levels in a hierarchically organized dimension, and alternatively can subcontract administration of
10 names below a given level of the dimension to registrants of the names on that level. Additionally, the authority charges usage fees for use of the name look-up service and the proximity-based search service. This usage fee generally is charged to the user of the client device 302 or of a service that employs the authoritative names registry for proximity assessment. Alternatively, the usage fee can be charged back to the registrant
15 (at any level) of a name upon access for either the name look-up service or proximity-based search service for a user.

 In particular with respect to the devices dimension of the names registry, the authority in the illustrated implementation provides generic device type names at initial levels (e.g., levels 392, 393) of the devices dimension in the authoritative names
20 registry, which are not registered to nor charged registration fees to any registrant. However, the authority registers specific device type names at a level (e.g., level 394) that appear under a generic device type name to registrants (such as a device manufacturer, and charges recurring (e.g., on an annual or other periodic basis) registration fees to the registrants. The registrant owns the specific device type name,
25 and provides the device-specific XML or COM object file on the computer network that is accessed with the name. The authority may also charge registration fees to the registrant for registering additional names at lower levels (e.g., 395-396) under such first device-specific name in the devices dimension hierarchy 380. Alternatively, the registrant owning a specific device type name can be permitted to register such lower

level device specific names without further charge, and provide its own names registry services for such lower level device specific names.

Having described and illustrated the principles of our invention with reference to an illustrated embodiment, it will be recognized that the illustrated embodiment can be
5 modified in arrangement and detail without departing from such principles. It should be understood that the programs, processes, or methods described herein are not related or limited to any particular type of computer apparatus, unless indicated otherwise. Various types of general purpose or specialized computer apparatus may be used with or perform operations in accordance with the teachings described herein. Elements of
10 the illustrated embodiment shown in software may be implemented in hardware and vice versa.

In view of the many possible embodiments to which the principles of our invention may be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of our invention. Rather,
15 we claim as our invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.

We claim:

1. A process of operating a names registry for access to device-specific protocols, the process comprising:
 - 5 providing a names registry having names organized in a hierarchical organization, the hierarchical organization having a top level name pertaining to a devices dimension of the names registry, and having a next lower level of names pertaining to generic device types;
 - registering specific device type names at levels of the hierarchy under the
 - 10 generic device type names to registrants;
 - associating at least some of the specific device type names to an networking address via which a device-specific protocol can be accessed for operationally interacting with a device of a type denoted by the respective specific device type name;
 - providing a name look-up service operative in response to a look-up request
 - 15 designating a device name from the names registry to return the networking address associated with the device name; and
 - assessing recurring registration fees to the registrants for registration of the specific device type names.
- 20 2. The process of claim 1 further comprising:
 - assessing a usage fee to a registrant of the device name for uses of the name look-up service in look-up request designating the device name.
3. The process of claim 1 further comprising:
 - 25 assessing a usage fee to a requester that issues the look-up request to the name look-up service.
4. The process of claim 1 further comprising:

conditioning access to the device-specific protocol using cryptographic security.

5. A process of operating a names registry for use in proximity networking in a pervasive computing environment, the process comprising:

5 providing a names registry having names organized in multiple dimensions, the dimensions representing at least some of devices, services, networks, venues, spaces, people, processes, and events, and the dimensions further defining a measure of proximity;

10 registering names in association with particular of the dimensions into the names registry to registrants;

providing a search service operative in response to a search query from a user of a client device to return links to devices or services based on proximity defined per the names registry; and

15 assessing recurring registration fees to the registrants for registration of the names.

6. The process of claim 5 wherein the search query has user-specified explicit search terms and names, the process further comprising:

20 associating names from the names registry that relate to use of the client device by the user as implicit search parameters to the search query;

associating sets of names from the names registry that relate to devices and services accessible within the pervasive computing environment to such devices and services; and

25 assessing proximity of user to the devices and services by calculating differences of the names associated as implicit search parameters with the search query from the sets of names associated with the devices and services.

7. The process of claim 6 wherein the search query is a request for an inverse search, the process further comprising:

responsive to such inverse search request, issuing an alert notification to the client device when devices or services meeting the explicit search term also meet a proximity criteria based on the implicit search parameters.

- 5 8. An authority-managed multiple-dimensional names registry residing on a collection of names service computers, the registry comprising:
- a collection of names in a devices dimension relating to embedded computing devices;
- a names mapping association of at least some of the names in the devices
- 10 dimension to computer network addresses with which device control protocols of the embedded computing devices are accessed; and
- a collection of names in at least one other dimension of a group including services, networks, venues, spaces, people, processes, and events.

FIG. 1

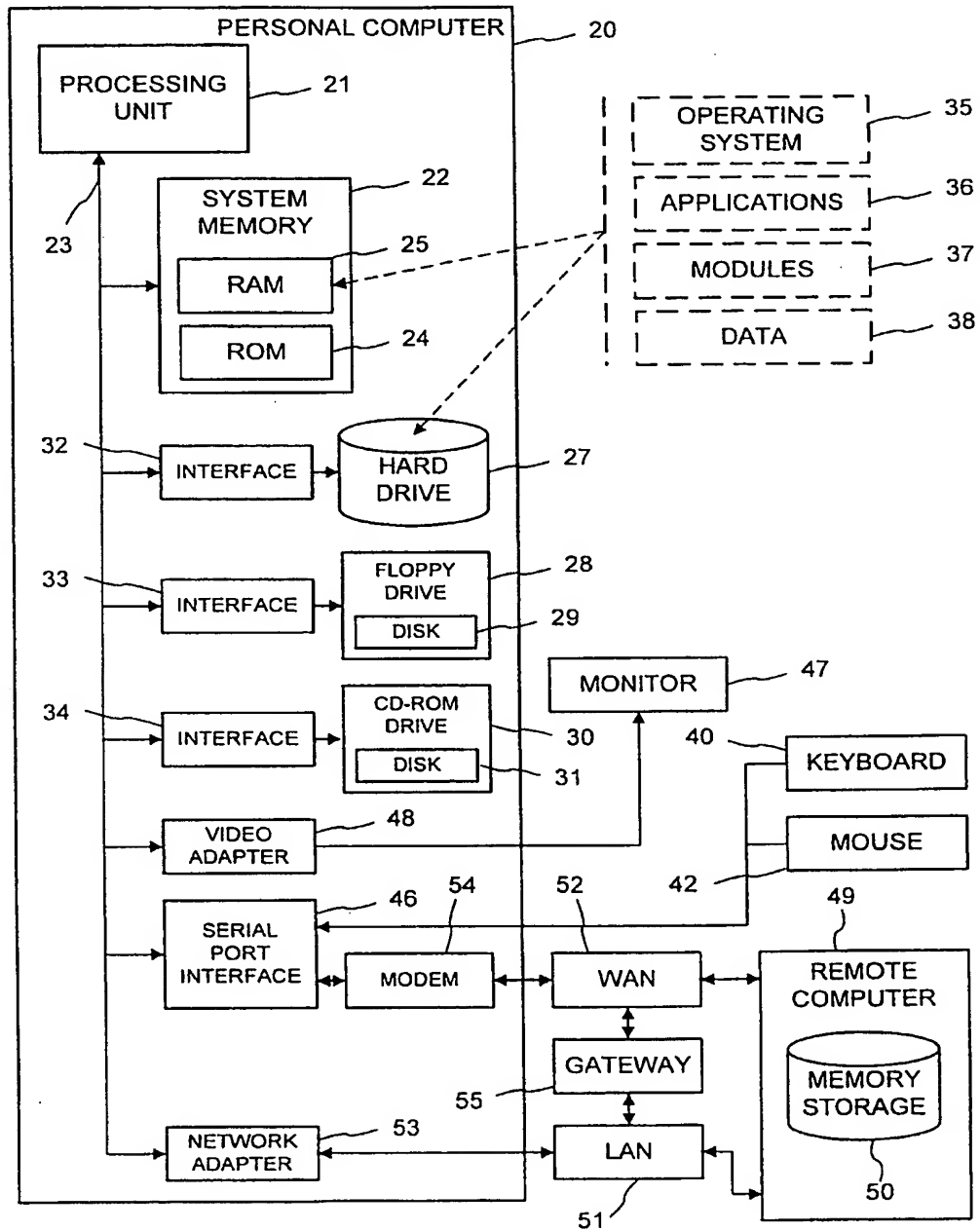


FIG. 2

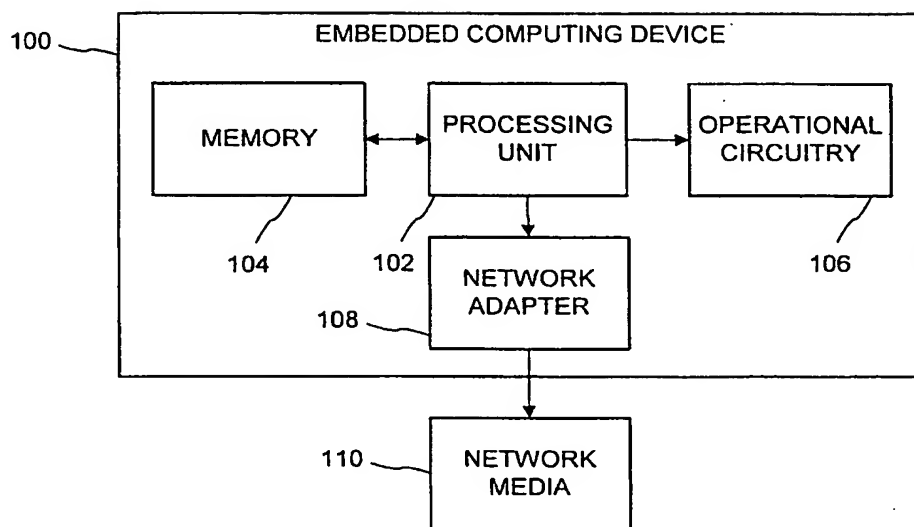


FIG. 3

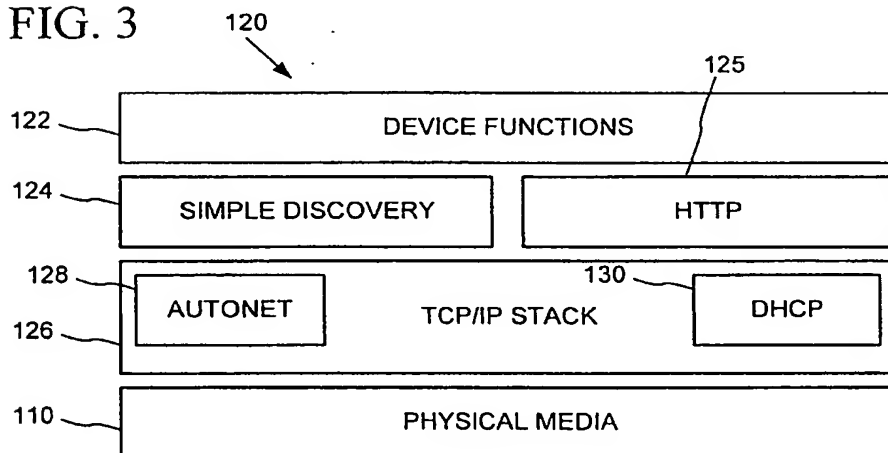


FIG. 4

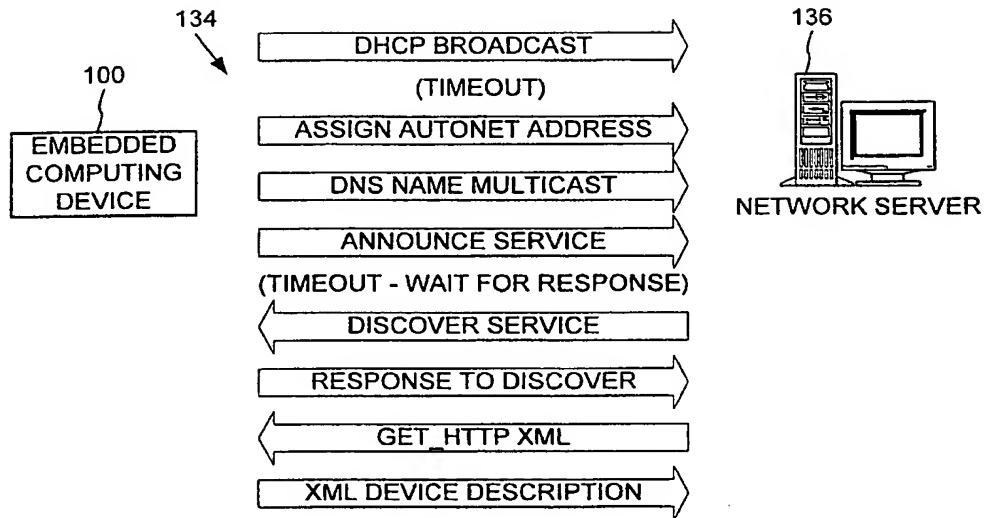
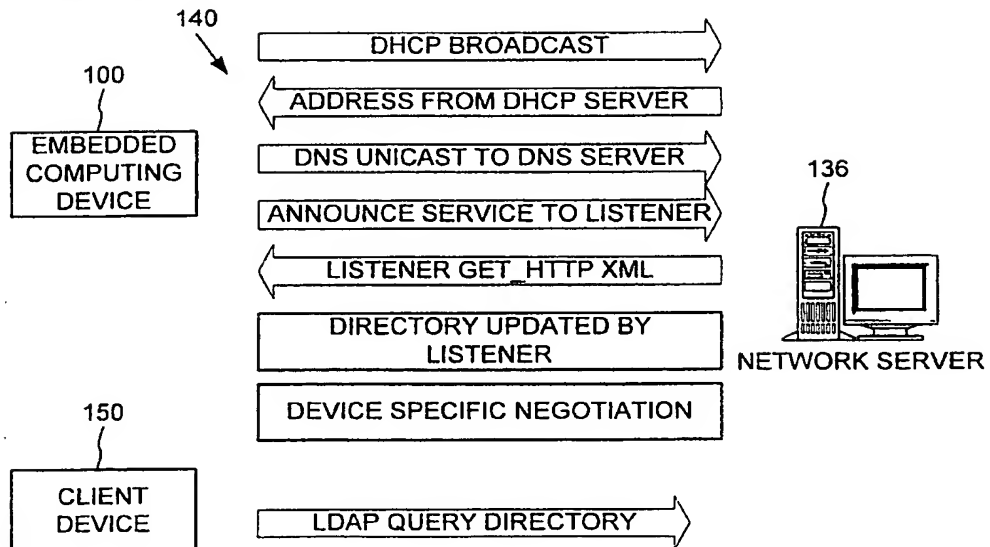


FIG. 5



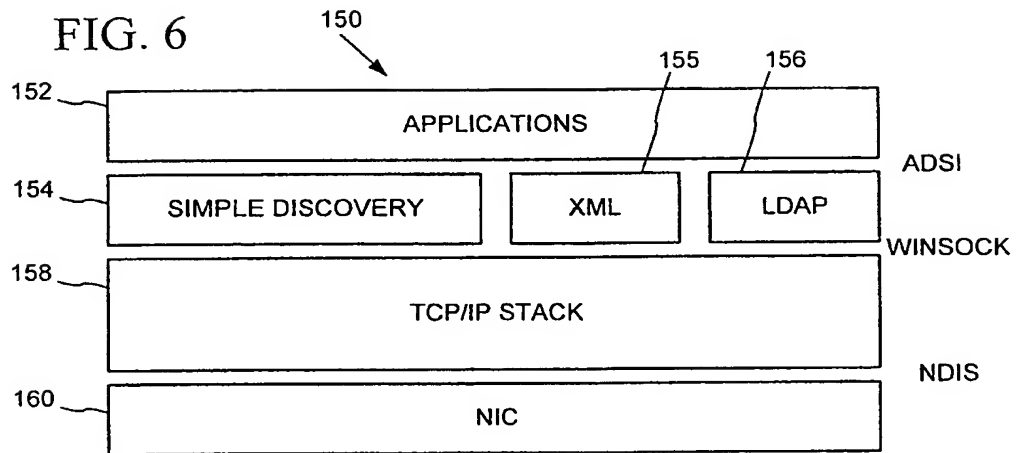


FIG. 7

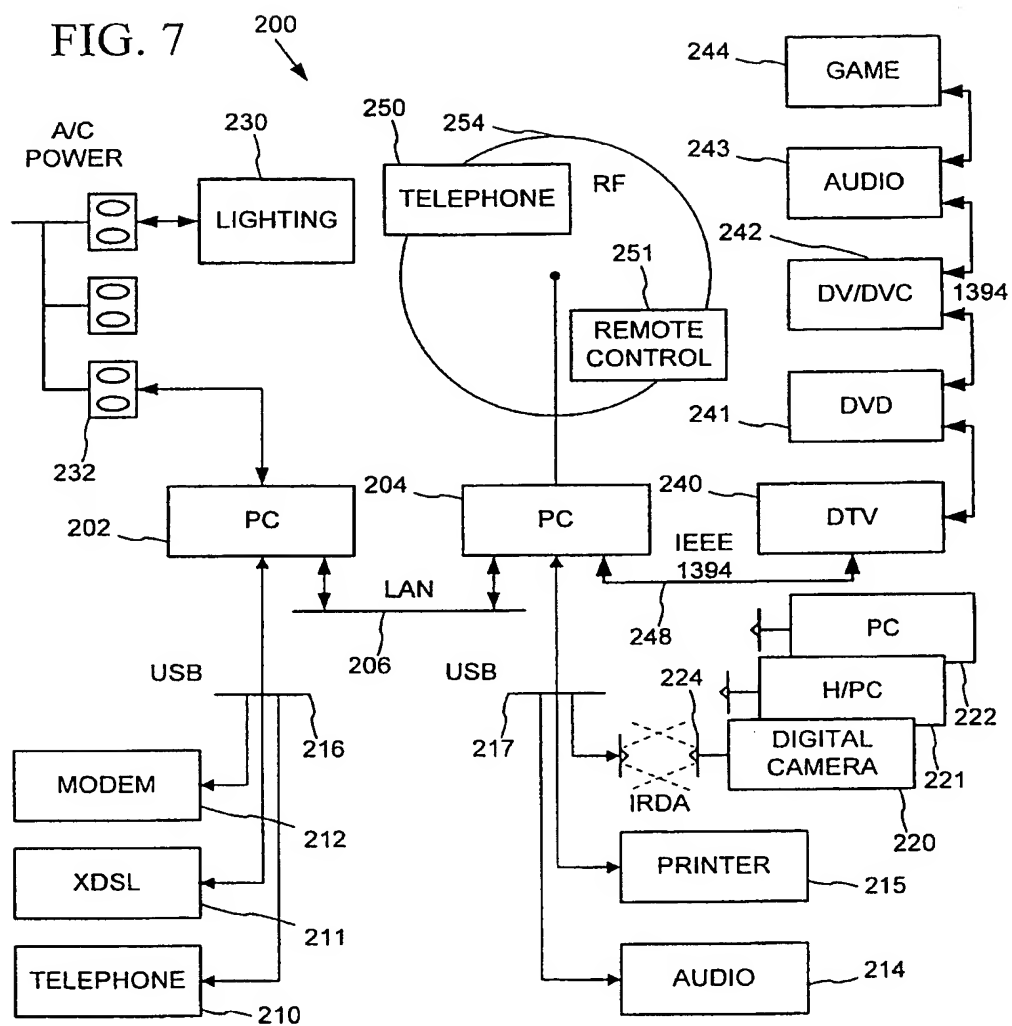


FIG. 8

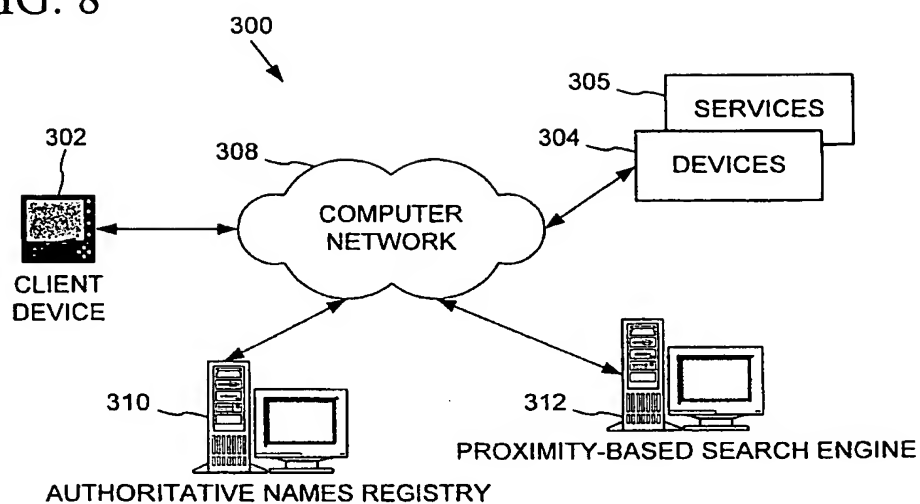


FIG. 9

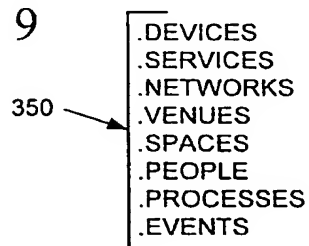


FIG. 10

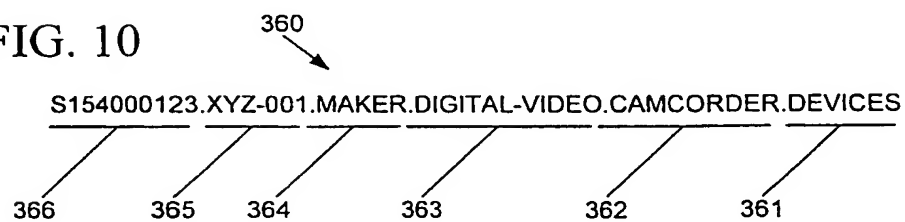


FIG. 11

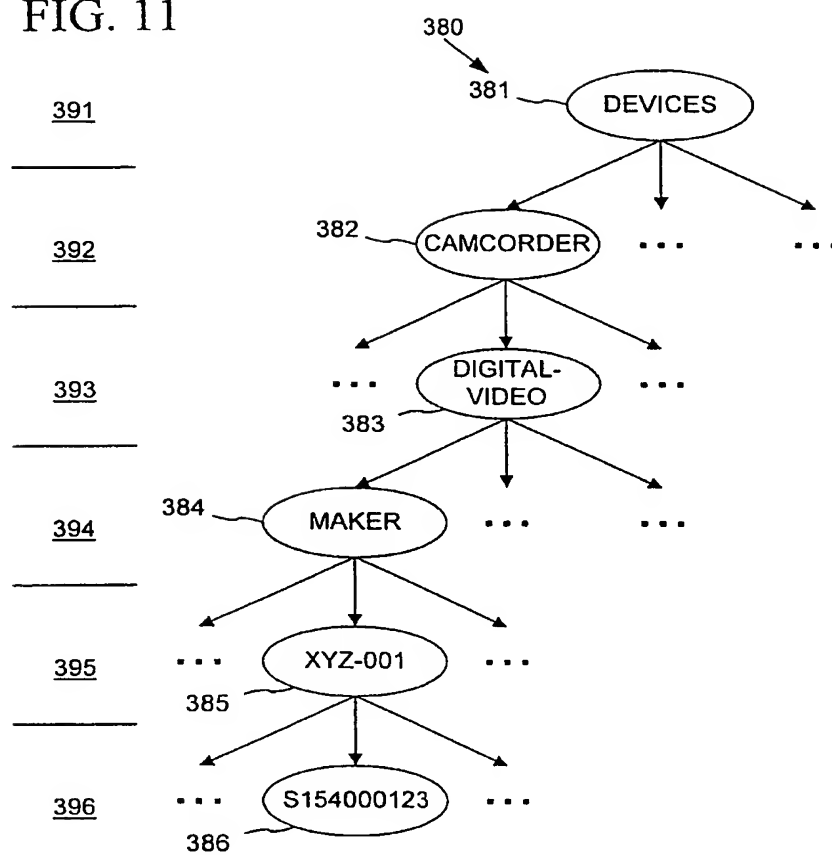


FIG. 12

400

← → START MAIL NEWS FAVORITES

SEARCH

404 406 GO

402 SEARCH IN

- ☒ NEIGHBORHOOD
- ☐ ARCHIVE
- ☐ HISTORY
- ☐ WEB
- ☐ ALL
- 408 ☐ INVERSE SEARCH

MODIFY SEARCH

MONDAY 12:34 PM